

Pogoplug configuration as an NFS server

George Jones

<2014-07-12 Sat>

Contents

1	[0/8] Install basic pogoplug	2
1.1	TODO Plug in pogoplug power	2
1.2	TODO Plug in pogoplug to Internet connected port	2
1.3	TODO Go to pogoplug.com/activate	2
1.4	TODO Type in 26 digit/number ID on bottom of unit if needed.	2
1.5	TODO Enable SSH on the pogoplug web site.	2
1.6	TODO Change the password	3
1.7	TODO Test SSH.	3
1.8	TODO [optional] Assign a fixed address to the pogoplug.	3
2	TODO Root the Pogoplug	3
3	TODO Reboot	3
4	TODO Attach storage	3
4.1	TODO install NTFS-3G to enable NTFS writing	3
4.2	TODO Find the partition	3
4.3	TODO Mount the ntfs partition	3
4.4	TODO fix NTFS filesystem problems	4
5	[0/3] Set up a permanent mount of the storage on the pogoplug	4
5.1	TODO Find the UUID	4
5.2	TODO Add a user	4
5.3	TODO create the fstab entry with linux compatible permissions allowing	5
6	TODO Install NTP	5

7	[0/7] TODO Install and Configure NFS	5
7.1	TODO Install NFS Utilities	5
7.2	TODO Set up idmapd.conf	5
7.3	TODO Set up NFS options	6
7.4	TODO (re)start the NFS server	6
7.5	TODO Export the filesystem	6
7.6	TODO Start NFS server on reboot	6
7.7	IN-PROGRESS Open up firewall rules if needed	7
8	[0/3] Client side NFS configuration	7
8.1	TODO Configure NTP on the client	7
8.2	TODO Mount from the client by hand (assumes Ubuntu with packages installed)	7
8.3	TODO Set up mount from fstab on boot	8
9	Outstanding Problems	8

This is a log of the steps that I took to install arch linux on a pogoplug 4 and set it up as a local storage device, export nfs and mount it. This is done for myself as a record of what I've done because 1) I might have to repeat the process and 2) someone else might find it useful.

Start here: <http://archlinuxarm.org/platforms/armv5/pogoplug-series-4>

1 [0/8] Install basic pogoplug

- 1.1 TODO Plug in pogoplug power**
- 1.2 TODO Plug in pogoplug to Internet connected port**
- 1.3 TODO Go to pogoplug.com/activate**
- 1.4 TODO Type in 26 digit/number ID on bottom of unit if needed.**
- 1.5 TODO Enable SSH on the pogoplug web site.**

See <https://pogoplug.com/settings>. Herein lies the risk should pogoplug go belly up, decide they don't want people rooting their devices anymore, etc.

1.6 TODO Change the password

1.7 TODO Test SSH.

Get address assigned by DHCP from devices attached to local router/WAP (in my case, <http://192.168.1.1>, login in as appropriate and look at connected devices to find pogoplug)

1.8 TODO [optional] Assign a fixed address to the pogoplug.

I did this by adding a DNS reservation at a fixed address for the MAC address of the Pogoplug.

2 TODO Root the Pogoplug

See <http://archlinuxarm.org/platforms/armv5/pogoplug-series-4>. I used a USB thumb drive in the top USB slot (you have to remove the cover)

3 TODO Reboot

4 TODO Attach storage

I attached a 2 TB external USB drive to one of the USB ports which is formatted with NTFS. I chose to keep it rather than formatting with ext4 so I can plug it in to other devices that understand NTFS.

4.1 TODO install NTFS-3G to enable NTFS writing

```
[root@alarm ~]# pacman -S ntfs-3g
```

4.2 TODO Find the partition

```
[root@alarm ~]# fdisk -l
```

4.3 TODO Mount the ntfs partition

```
[root@alarm ~]# mount -t ntfs-3g /dev/sdb1 /mnt/data
```

I got the following error, because, apparently the partition was not correctly dismounted:

```
[root@alarm ~]# mount -t ntfs-3g /dev/sdb1 /mnt/data
mount -t ntfs-3g /dev/sdb1 /mnt/data
$MFTMirr does not match $MFT (record 0).
Failed to mount '/dev/sdb1': Input/output error
NTFS is either inconsistent, or there is a hardware fault, or it's a
SoftRAID/FakeRAID hardware. In the first case run chkdsk /f on Windows
then reboot into Windows twice. The usage of the /f parameter is very
important! If the device is a SoftRAID/FakeRAID then first activate
it and mount a different device under the /dev/mapper/ directory, (e.g.
/dev/mapper/nvidia_eahaabcc1). Please see the 'dmraid' documentation
for more details.
```

4.4 TODO fix NTFS filesystem problems

Given that I don't have a windows system to run chkdisk with, I'm considering reformatting with ext4 or similar and moving on... but some quick googling found an answer: <http://askubuntu.com/questions/47700/fix-corrupt-ntfs-partition->

```
[root@alarm ~]# ntfsfix /dev/sdb1
```

after which

```
mount -t ntfs-3g /dev/sdb1 /mnt/data
```

works just fine.

5 [0/3] Set up a permanent mount of the storage on the pogoplug

5.1 TODO Find the UUID

```
[root@alarm ~]# blkid /dev/sdb1
blkid /dev/sdb1
/dev/sdb1: LABEL="Seagate Expansion Drive" UUID="54F8DF61F8DF3FC2" TYPE="ntfs" PARTUUID="1"
```

5.2 TODO Add a user

```
useradd -m -g users -G wheel -s /bin/bash user
```

5.3 TODO create the fstab entry with linux compatible permissions allowing

```
cat <<END >> /etc/fstab
# Mount internal Windows partition with linux compatible permissions, i.e. 755 for dir
UUID=54F8DF61F8DF3FC2 /mnt/data ntfs-3g uid=user,gid=users,dmask=022,fmask=133 0 0
END
```

6 TODO Install NTP

See https://wiki.archlinux.org/index.php/Network_Time_Protocol_daemon#Installation

```
pacman -S ntp
ntpd -q
systemctl enable ntpd.service
systemctl start ntpd.service
ntpq -p
```

7 [0/7] TODO Install and Configure NFS

See <https://wiki.archlinux.org/index.php/NFS>

7.1 TODO Install NFS Utilities

```
pacman -S nfs-utils
```

7.2 TODO Set up idmapd.conf

```
cat <<HERE > /etc/idmapd.conf
[General]

Verbosity = 1
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = atomic

[Mapping]

Nobody-User = nobody
Nobody-Group = nobody
```

```
[General]
HERE
```

7.3 TODO Set up NFS options

```
if [ ! -f /etc/conf.d/nfs-common.conf.orig ]; then
    cp /etc/conf.d/nfs-common.conf /etc/conf.d/nfs-common.conf.orig
fi
```

```
cat /etc/conf.d/nfs-common.conf.orig | sed -e 's/^STATD_OPTS.*/STATD_OPTS="-p 32765 -o
```

```
if [ ! -f /etc/conf.d/nfs-server.conf.orig ]; then
    cp /etc/conf.d/nfs-server.conf /etc/conf.d/nfs-server.conf.orig
fi
```

```
cat /etc/conf.d/nfs-server.conf.orig | sed -e 's/^MOUNTD_OPTS.*/MOUNTD_OPTS="-p 20048"
```

7.4 TODO (re)start the NFS server

```
systemctl restart nfs-config
systemctl stop rpc-statd
systemctl start rpc-statd
systemctl stop nfs-server
systemctl start nfs-server
rpcinfo -p
```

7.5 TODO Export the filesystem

```
if [ ! -f /etc/exports.orig ]; then
    cp /etc/exports /etc/exports.orig
fi
```

```
cp /etc/exports.orig /etc/exports
echo "/mnt/data 192.168.1.0/24(rw,no_subtree_check,nohide,all_squash,anonuid=0,anongid=0)
```

```
exportfs -rav
```

7.6 TODO Start NFS server on reboot

```
systemctl enable nfs-server.service
```

```
systemctl start nfs-server.service
```

7.7 IN-PROGRESS Open up firewall rules if needed

```
if [ ! -f /etc/iptables/iptables.rules.orig ]; then
    cp /etc/iptables/iptables.rules /etc/iptables/iptables.rules.orig
fi
```

```
cp /etc/iptables/iptables.rules.orig /etc/iptables/iptables.rules
```

```
cat <<HERE >> /etc/iptables/iptables.rules
-A INPUT -p tcp -m tcp --dport 111 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 2049 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 20048 -j ACCEPT
-A INPUT -p udp -m udp --dport 111 -j ACCEPT
-A INPUT -p udp -m udp --dport 2049 -j ACCEPT
-A INPUT -p udp -m udp --dport 20048 -j ACCEPT
```

If using NFSv3 and the above listed static ports for rpc.statd and lockd these also ne

```
/etc/iptables/iptables.rules
-A INPUT -p tcp -m tcp --dport 32765 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 32803 -j ACCEPT
-A INPUT -p udp -m udp --dport 32765 -j ACCEPT
-A INPUT -p udp -m udp --dport 32803 -j ACCEPT
HERE
```

8 [0/3] Client side NFS configuration

8.1 TODO Configure NTP on the client

Left as an exercise for the reader. Your clients may vary.

8.2 TODO Mount from the client by hand (assumes Ubuntu with packages installed)

```
showmount -e 192.168.1.222
mount -t nfs 192.168.1.222:/mnt/data /mnt/data
df
```

8.3 TODO Set up mount from fstab on boot

```
if [ ! -f /etc/fstab.orig ]; then
    cp /etc/fstab /etc/fstab.orig
fi

cp /etc/fstab.orig /etc/fstab

cat <<HERE >> /etc/fstab
192.168.1.222:/mnt/data    /mnt/data    nfs4    rsize=8192,wsiz=8192,timeo=14,_netdev
0 0
HERE
```

9 Outstanding Problems

- None.